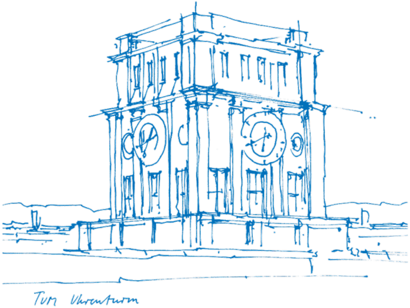
Analysis of Machine Learning for State Register Identification

Lee Seng Hwee
[1] Technische Universität München
Faculty of Electrical and Computer Engineering
Institute for Security in Information Technology

University
July 2, 2020



Outline

Introduction

Limitation of Current Approaches

Traditional approach:

- Requires golden model

RELIC and fastRELIC:

- Based solely on Pair Similarity Score(PSS)

Problem Statement

Reliance on a golden model in the traditional method results in severe identification limitations, while RELIC/fastRELIC which relied solely on PSS, results in a lower accuracy.

Proposed Solution

To train and deploy a machine learning model for State Register identification.

Advantages of Machine Learning:

- Faster Processing
- Ease of use
- Consistency in evaluation
- Rely on multiple features

Implementations

The Data

Prior to creating a Neural Network for State Register Identification, 4 methods of implementating the feature file was discussed for training the Neural Network

- Original features set
- Original features set with Euclidean Distance Similarity Score
- Original features set with fastRELIC Similarity Score
- Original features set with Euclidean and fastRELIC Similarity Score

The Original Features

Average Neighbour Degree	Betweenness Centrality
Closeness Centrality	Clustering
Degree	Degree Centrality
Indegree	Has Feedback Path
Katz	Load Centrality
Outdegree	Pagerank

Table: Original Features

Euclidean Distance Similarity Score

- By extracting Register Shapes from design files

```
'DFFPOSX1_2': ['DFFPOSX1',
               'INPUT',
               'OR2X2',
               'AND2X2',
               'DFFPOSX1',
               'OR2X2',
               'OR2X2',
               'AND2X2',
               'DFFPOSX1',
               'OR2X2',
               'AND2X2',
               'AND2X2',
               'INVX1',
               'INVX1',
               'INPUT'],
```

```
{'DFFPOSX1_1': [3, 2, 3, 4, 3],
 'DFFPOSX1_2': [4, 3, 2, 2, 4],
 'DFFPOSX1_3': [3, 4, 2, 2, 1],
 'DFFPOSX1_4': [3, 2, 2, 1, 4]}
```

Euclidean Distance Similarity Score

- Comparing vectors, producing a similarity score using Euclidean Distance

$$E(u, v) = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}$$

where u and v are n^{th}-dimensional vectors

- Count++, if $E(u, v) < \mathit{Threshold}$
- Normalize

fastRELIC Similarity Score

- Using PSS algorithm, similarity score between 0 to 1
- Count++, if $PSS > Threshold$
- Normalize

Features Selection

Constant Filtering

- Removing features with constant values

Quasi-constant filtering

- *Removing features with a value difference less than a selected threshold*

Feature Permutation

- Permutes the values in a feature
- Trains data set with the permuted feature on pre-optimised neural network
- Compare permuted accuracy with unpermuted accuracy

Sequential Feature Selection

- Train on pre-optimised neural network
- Select feature combinations based on accuracy by adding features one at a time

Methodology and Results

Testing Implementaion (Pre-Optimize Neural Network)

- Rotation of 13 files, 12 for training, 1 for testing
  - Train with files B – N, test with file A
  - Train with files A, C – N, test with file B
- Per file was used to train the model 100 times, experiment repeated 5 times
- Average result per implementation across all 13 tests

$$\text{A} = \frac{\text{Number of Correctly Predicted Registers}}{\text{Total Number of Registers}}$$

$$\text{SRA} = \frac{\text{Number of Correctly Predicted State Registers}}{\text{Total Number of State Registers}}$$

Results for different implementation

Implementation	Mean Model Acc	Mean State Register Acc
Original	0.75	0.59
With fastRELIC	0.86	0.77
With Euclidean	0.83	0.71
With Euclidean and fastRELIC	0.87	0.78

Table: Accuracy

Feature Permutation Methodology

- Rotation of 13 files, 12 for training, 1 for testing
- Each feature in a file permuted individually and train the model for 100 times
- Average the 100 accuracies per features
- Train model with non-permuted data set for 100 times and calculate average
- Repeat experiment for 5 times
- Calculate Ratio(Method 1) and Feature Occurrence(Method 2)

Ratio — Method 1

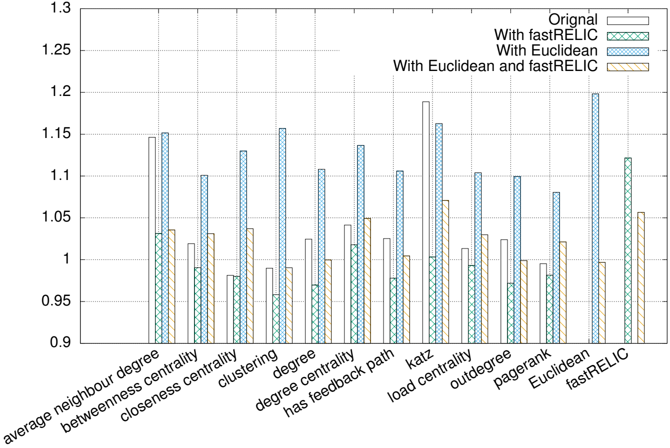
$$R_n(A_{original}, A_{permuted}) = \frac{A_{original}}{A_{permuted}}$$

$$SRR_n(SRA_{original}, SRA_{permuted}) = \frac{SRA_{original}}{SRA_{permuted}}$$

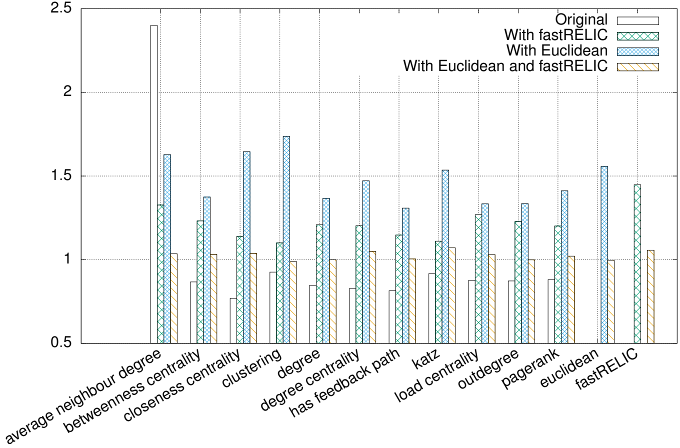
$R_n < 1$	$SRR_n < 1$	Feature Hindrance
$R_n = 1$	$SRR_n = 1$	Feature Hindrance
$R_n > 1$	$SRR_n > 1$	Feature Important

Table: Ratio Interpretation

Feature Permutation Method 1
Model Accuracy Ratio



Feature Permutation Method 1
State Register Accuracy Ratio



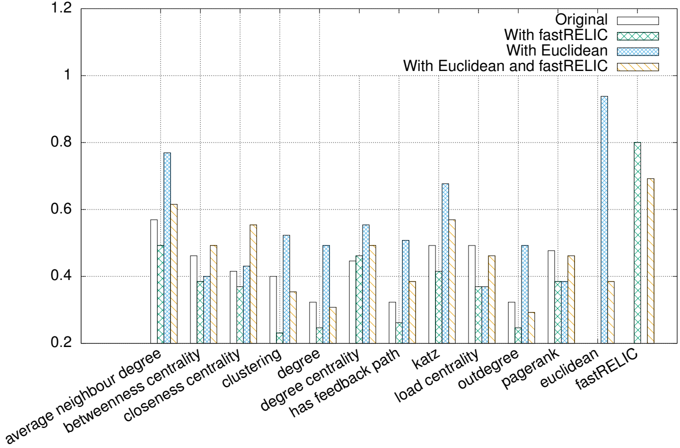
Feature Occurrence — Method 2

- Removing features using Table 4
- Count total number of times feature appear across all implementation per file
- Average Count and Normalize

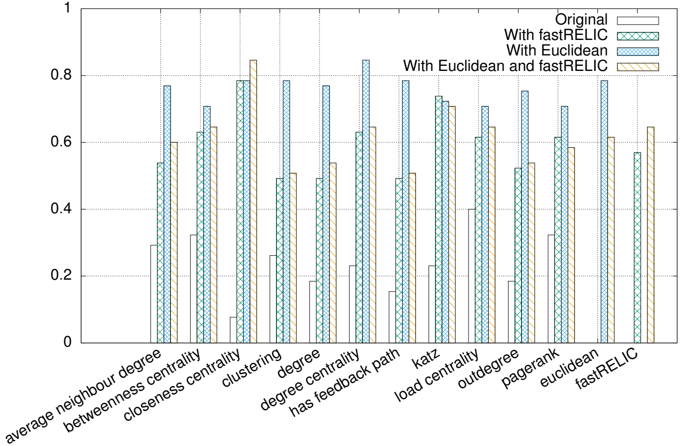
$A_{original} < A_{permuted}$	Feature Hindrance
$A_{original} = A_{permuted}$	Feature Hindrance
$A_{original} > A_{permuted}$, with a difference of $> 1\%$	Feature Important

Table: Conditions for filtering

Feature Permutation Method 2
Model Feature Occurrence



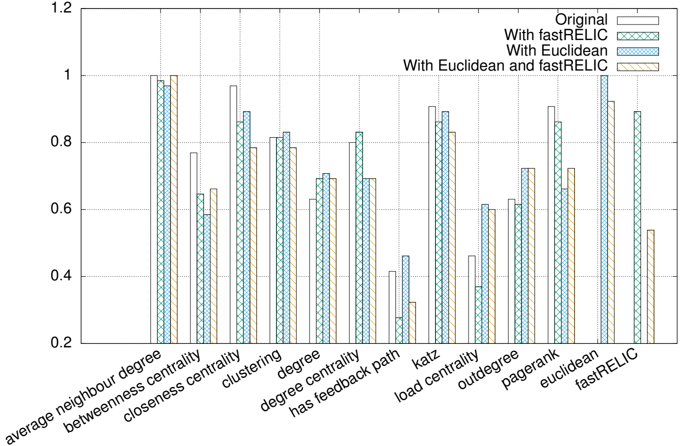
Feature Permutation Method 2
State Register Feature Occurrence



Sequential Feature Selection

- Rotation of 13 files, 12 for training, 1 for testing
- Run 5 times per file per implementation
- Count occurence per feature across all files
- Count total number of times feature appear across all implementation per file
- Average Count and Normalize
- Discard Feature Occuring $< 50\%$

Sequential Feature Selection



Conclusion for feature selection

Important Features

- Average Neighbour Degree
- Katz

Redundant Features

- Has Feedback Path
- Load Centrality

Results after Removing Features

Implementation	Mean Model Acc	Mean State Register Acc
Original	0.75	0.59
With fastRELIC	0.86	0.77
With Euclidean	0.83	0.70
With Euclidean and fastRELIC	0.87	0.77

Table: Model Accuracy

Optimizing

Optimized Model Accuracy

Mean Model Acc	Mean State Register Acc
0.65	0.91

Table: Optimized Model Accuracy

- Low Model Accuracy — High False Positive
- Model is Underfitted — Epoch might be too low

Further tuning the Optimized Model

- Increasing the Epochs from 10 to 22

Mean Model Acc	Mean State Register Acc
0.75	0.91

Table: Modified Optimized Model Accuracy

Future Work

- Using a larger data set
- Studying feature Correlation
- Using other feature selection method, eg. exhaustive feature selection
- Removing dependency on registers per file
- Experimenting with other Neural Network Architecture and hyperparameters

Thank You

Register shape